

Classificazione Dataset

FoodX-251

Erba Sandro - 856327
Gaviraghi Elia - 869493

Pulizia del Training Set





- **Assunzioni**

- Distribuzione errori
- Correttezza test set

- **Ipotesi**

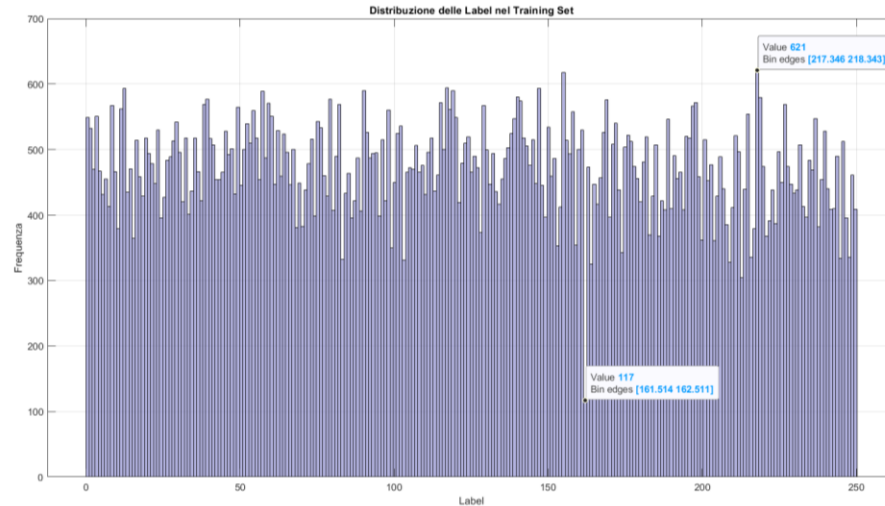
- Rete neurale addestrata sul test
- Clustering con distanza centroidi



Studio iniziale

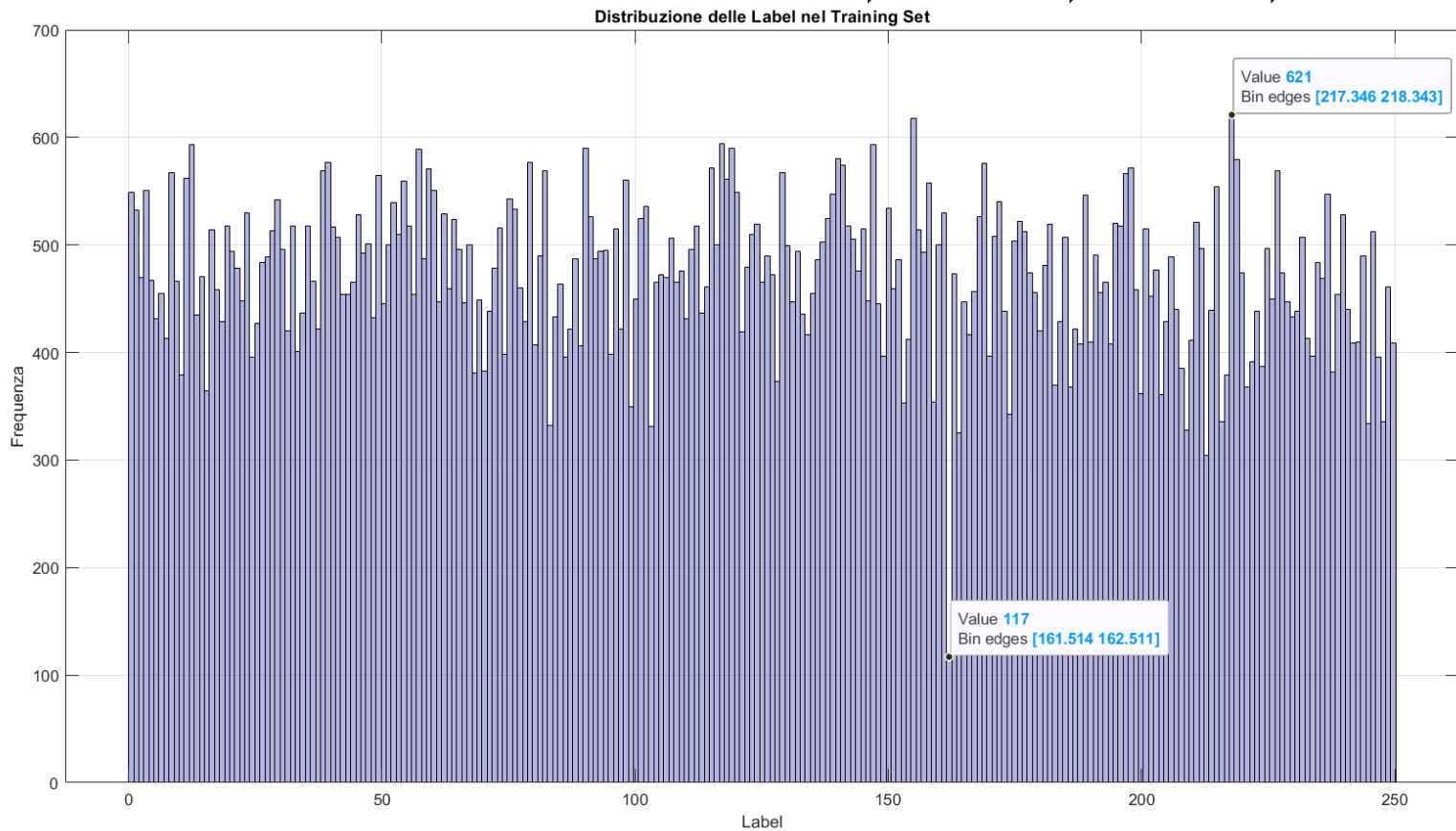
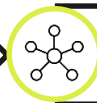


- Identificazione delle tipologie di errore
- Folderizzazione del train set
- Grafici di numerosità per monitorare istanze per classe

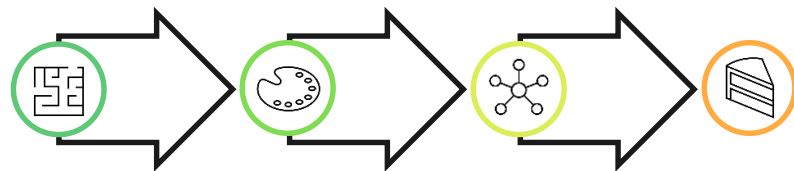




Studio iniziale



Ricerca in letteratura



- [FoodX-251: A Dataset for Fine-grained Food Classification](#): Introduce il dataset e classifica con una ResNet-101
- [An Artificial Intelligence-Based System to Assess Nutrient Intake for Hospitalised Patients](#): Classifica con una ResNet50 e sfrutta una GAN per data augmentation
- Libreria [FastDup](#) per riconoscimento degli outliers con cluster
- [iFood - 2019 at FGVC6](#) challenge in cui si usa questo dataset



Alexnet invertita

- **Idea**

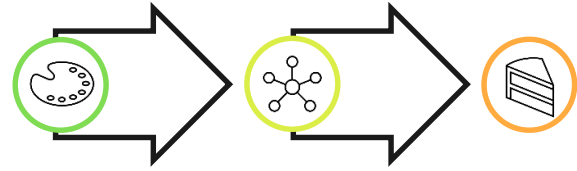
- Fine-tuning di una CNN pre-addestrata usando il test set

- **Dubbi iniziali**

- Rispetto delle richieste del progetto
- Effettivi risultati

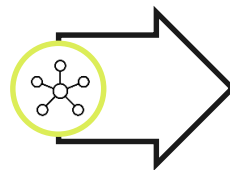
- **Problemi riscontrati**

- Enormi tempi di computazione
- Scarsi risultati iniziali





Media e varianza



- **Idea**
 - Calcolare media e varianza eliminando il 20% per ogni classe
- **Pro**
 - Rapido
 - Buona efficacia iniziale
- **Contro**
 - Feature non rappresentative per l'intero training set
 - Richiede di settare le soglie a mano

Macaron:



Non è un
macaron:



Potrebbero essere dei
macaron:





Clusterizzazione



- **Idea**

- Rafforzamento dei descrittori
- Clustering con i descrittori

- **Feature utilizzate**

- Feature extraction con una CNN pre-addestrata
- Rilevamento oggetti con HOG
- Texture con GLCM
- Media e varianza

- **Pro**

- Buona discriminazione degli outliers
- Metodo semplice

- **Contro**

- Ambiente di sviluppo e tempo computazionale richiesto

- **Ipotesi e scelte**

- Clustering classe per classe, feature per feature
- Taglio degli outliers recall oriented



Marble cake

- Da 117 a 24 immagini
- **Idea iniziale:** merging con classe 119 “coffee cake”
 - Pro: soluzione rapida e efficace
 - Contro: non valida ai fini del progetto
- **Soluzione scelta:** pulizia manuale e data augmentation
 - Riflessione
 - Rotazione
 - Traslazione
 - Da 24 a 192





Marble cake



Riflessione



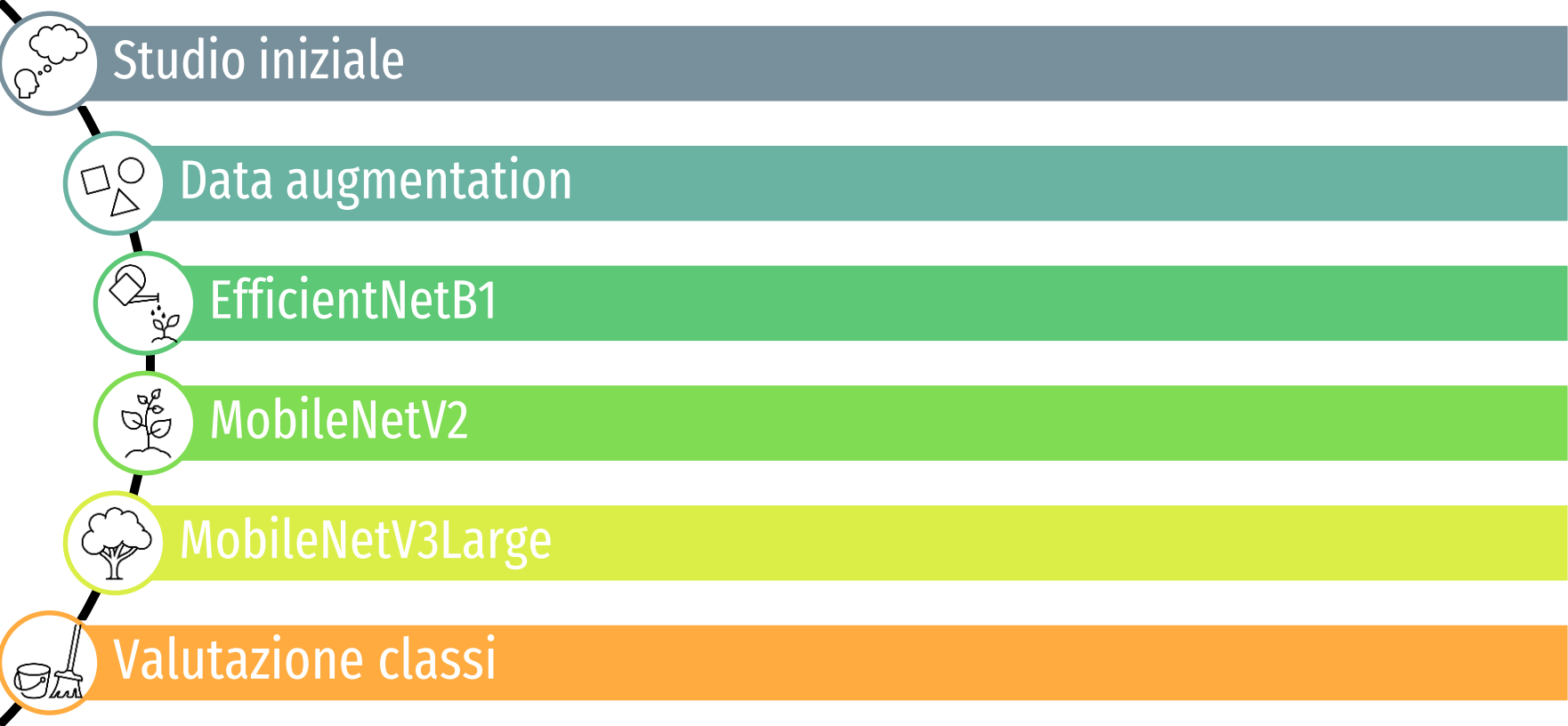
Rotazione



Traslazione

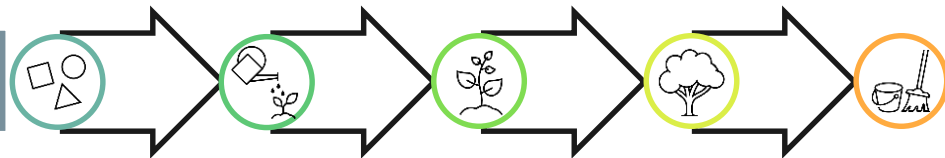


Addestramento CNN





Studio iniziale



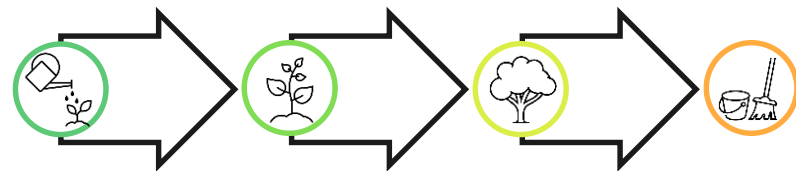
- **Problemi**

- GPU necessaria → Google Colab
- Risorse limitate → Scelta di *modelli leggeri* obbligata

- **Considerazioni**

- CNN da zero vs CNN pre-addestrate su ImageNet
 - “[...] in practice, very few people train an entire CNN from scratch, because it is relatively rare to have a dataset of sufficient size. It is common to pretrain a CNN on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the CNN either for fine-tuning or as a fixed feature extractor for the task of interest.” - Simone Bianco, CNN famous architecture, Advanced Machine Learning.
- Feature extraction vs fine-tuning
 - “[...] New dataset is small and similar to original dataset? train a linear classifier on CNN features from higher layers. New dataset is large and similar to original dataset? Fine-tune the CNN. New dataset is small but very different from original dataset? Train a linear classifier on CNN features from lower layers. New dataset is large and very different from original dataset? Train CNN from scratch or fine-tune it” - Simone Bianco, CNN famous architecture, Advanced Machine Learning.

Data augmentation

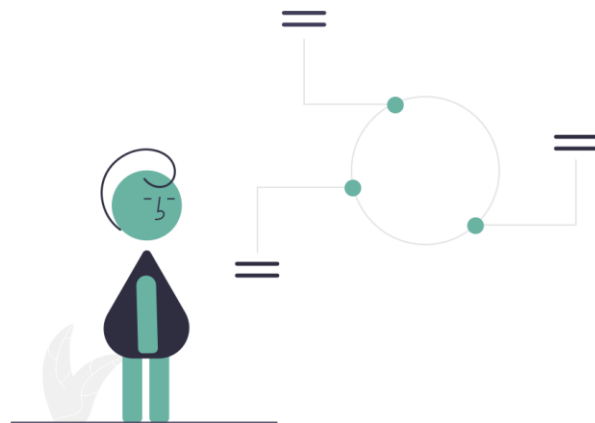


● Problemi

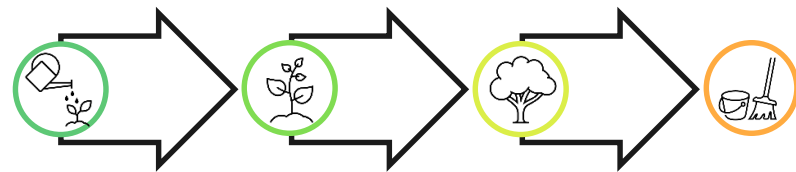
- Richieste del progetto
 - *Addestramento del classificatore per trattare tramite data augmentation anche le immagini a bassa qualità*
- Ipotesi applicative nella realtà

● Soluzioni

- Online batch-training
- Analisi test set degradato
- [DALib](#) e [Albumentations](#)



Data augmentation



Rotazione

Flip orizzontale

Prospettiva

Originale



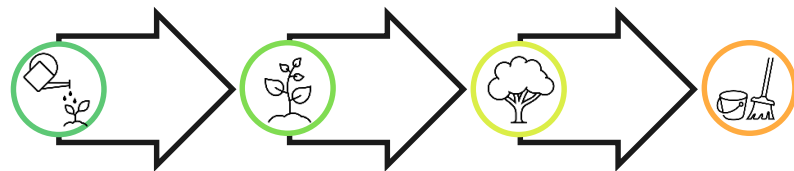
Blur

Noise gaussiano

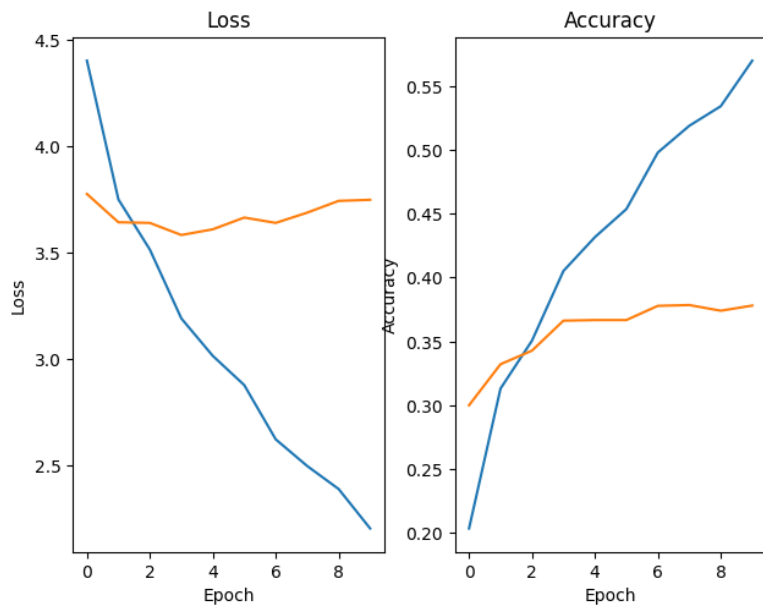
Compressione



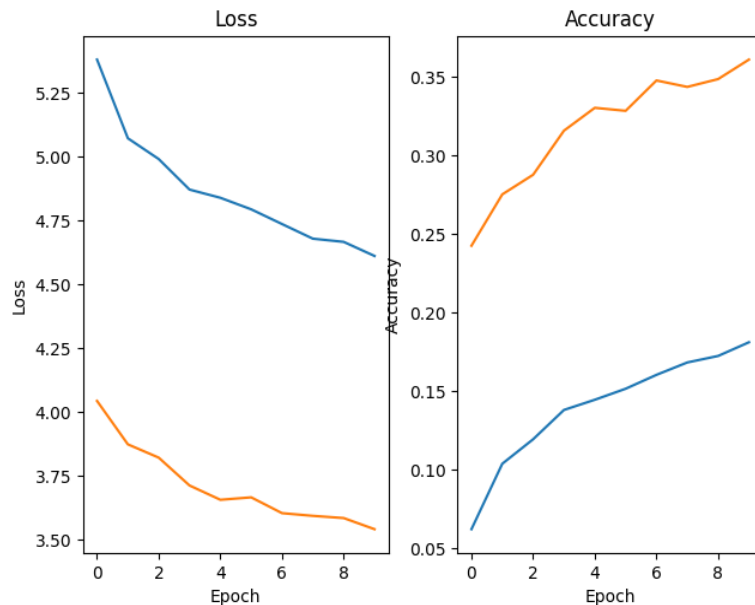
Data augmentation



Train senza data augmentation

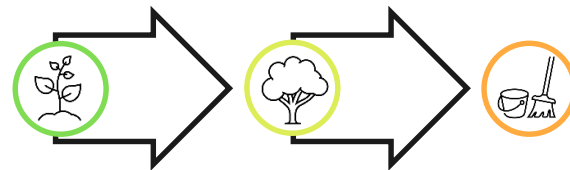


Train con data augmentation





EfficientNetB1

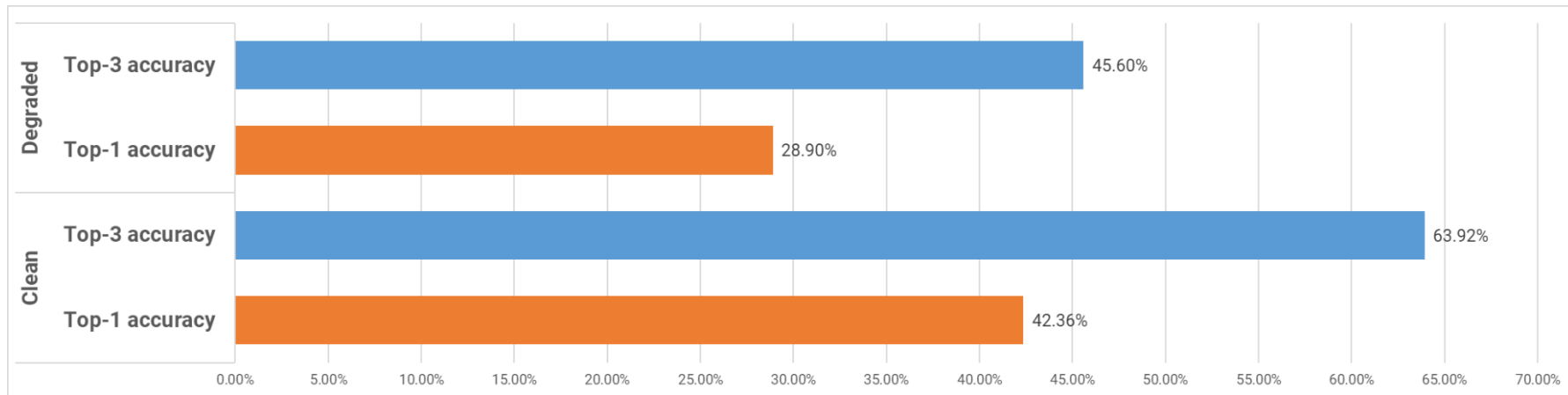


- Scelta

- fase di exploration dei modelli
- Risultati iniziali promettenti

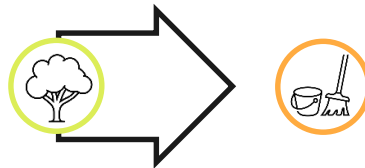
- Modello leggermente pesante

- vRAM GPU limitata
- Crash runtime con train prolungati





MobileNetV2

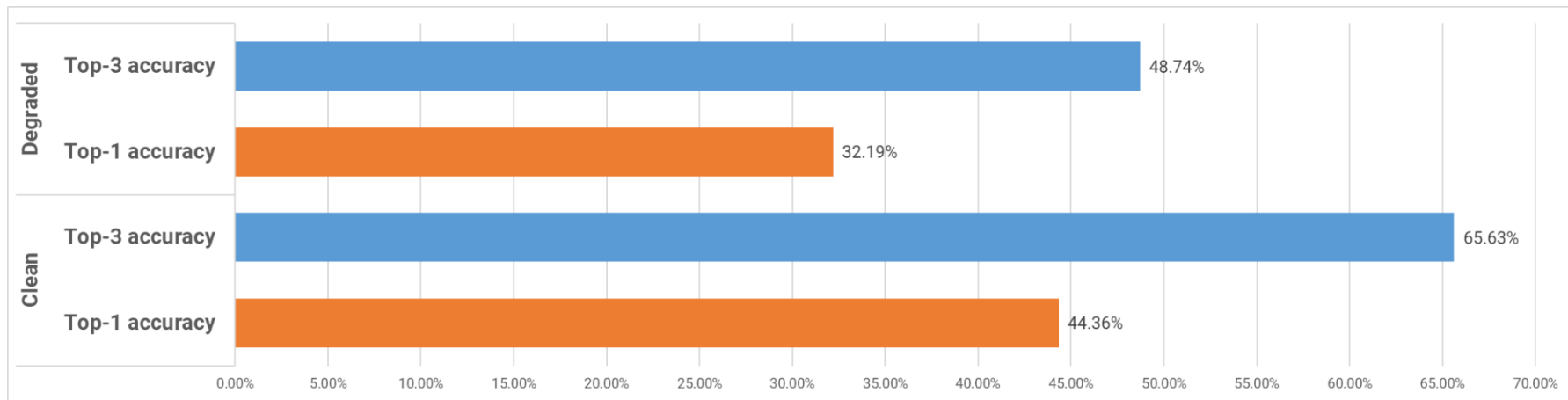


- **Scelta**

- Solo 3.5M di parametri
- Poca vRAM occupata, train libero

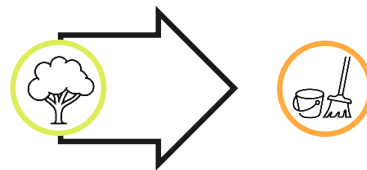
- Fase decisiva di *Exploration* di

- [Fine-tuning](#)
- Esplorazione degli iperparametri

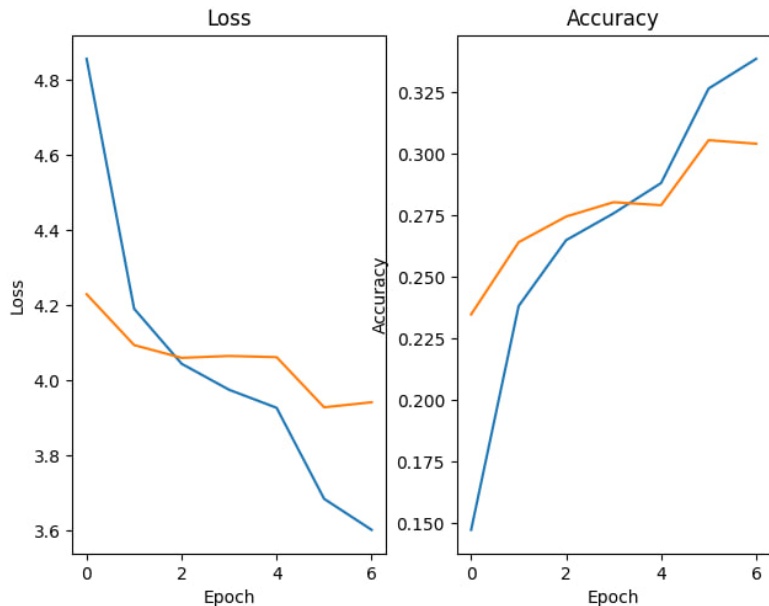




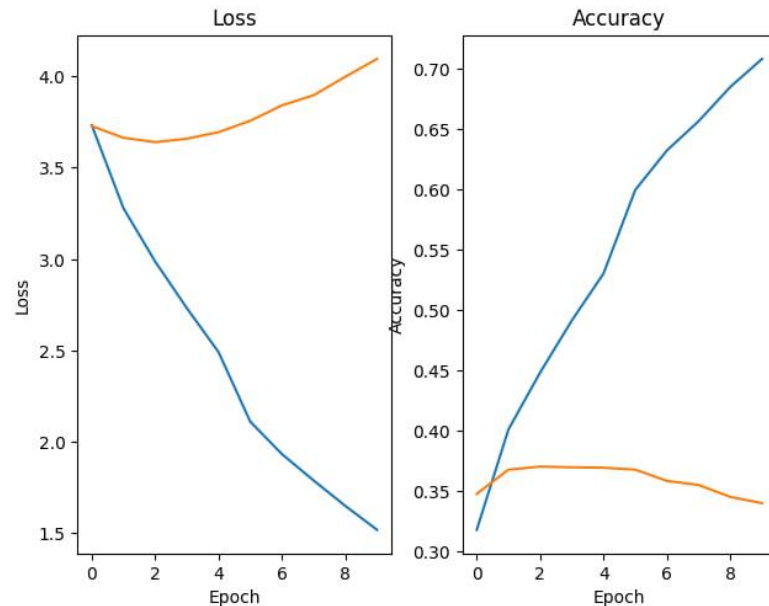
MobileNetV2



Andamento pre-train



Andamento defreeze-train





MobileNetV3Large

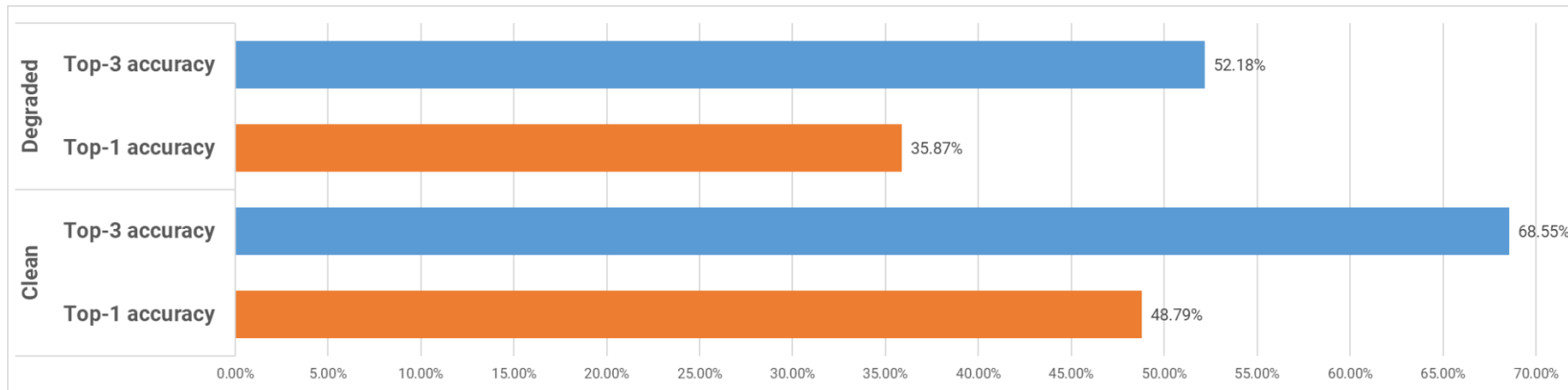


- **Scelta finale**

- Parametri: 5.5M vs 3.5M
- [ImageNet top-1 accuracy](#): 75.2% vs 71.3%

- **Fase decisiva di *Exploitation***

- **Modalità di fine-tuning**
- Esplorazione degli iperparametri





Valutazione classi

- **Problema**

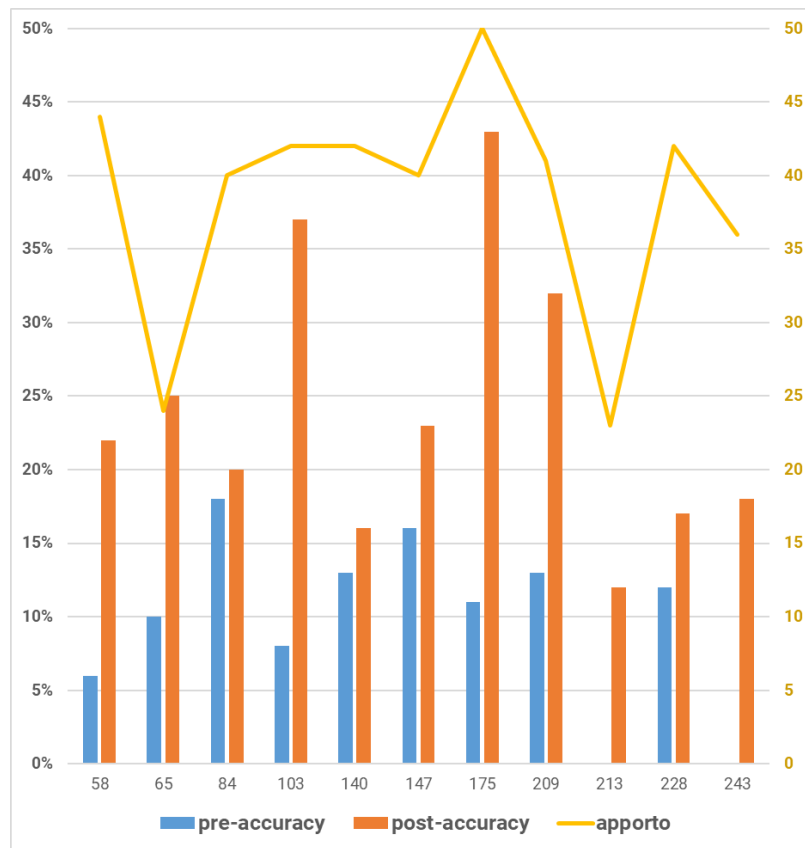
- Perché si passa da un accuracy top-1 dell'80% fino allo 0% in alcune classi?

- **Analisi**

- Visualizzazione delle classi con peggiori performance
- Valutazione possibile incisività

- **Risultato**

- Intervento manuale necessario
- Apporto al cambiamento

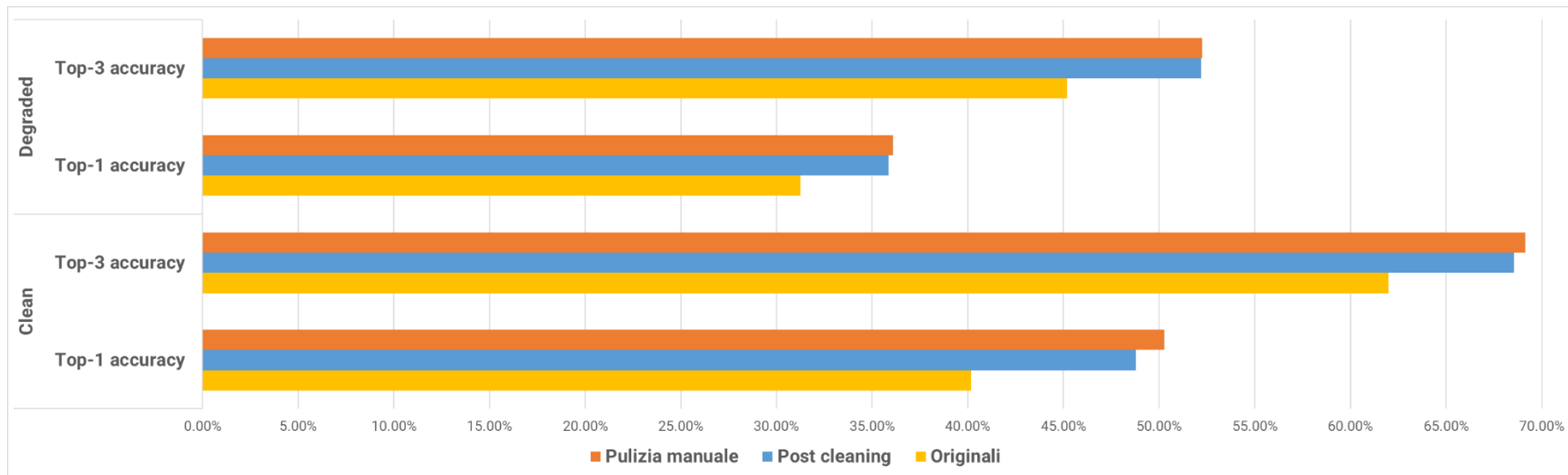




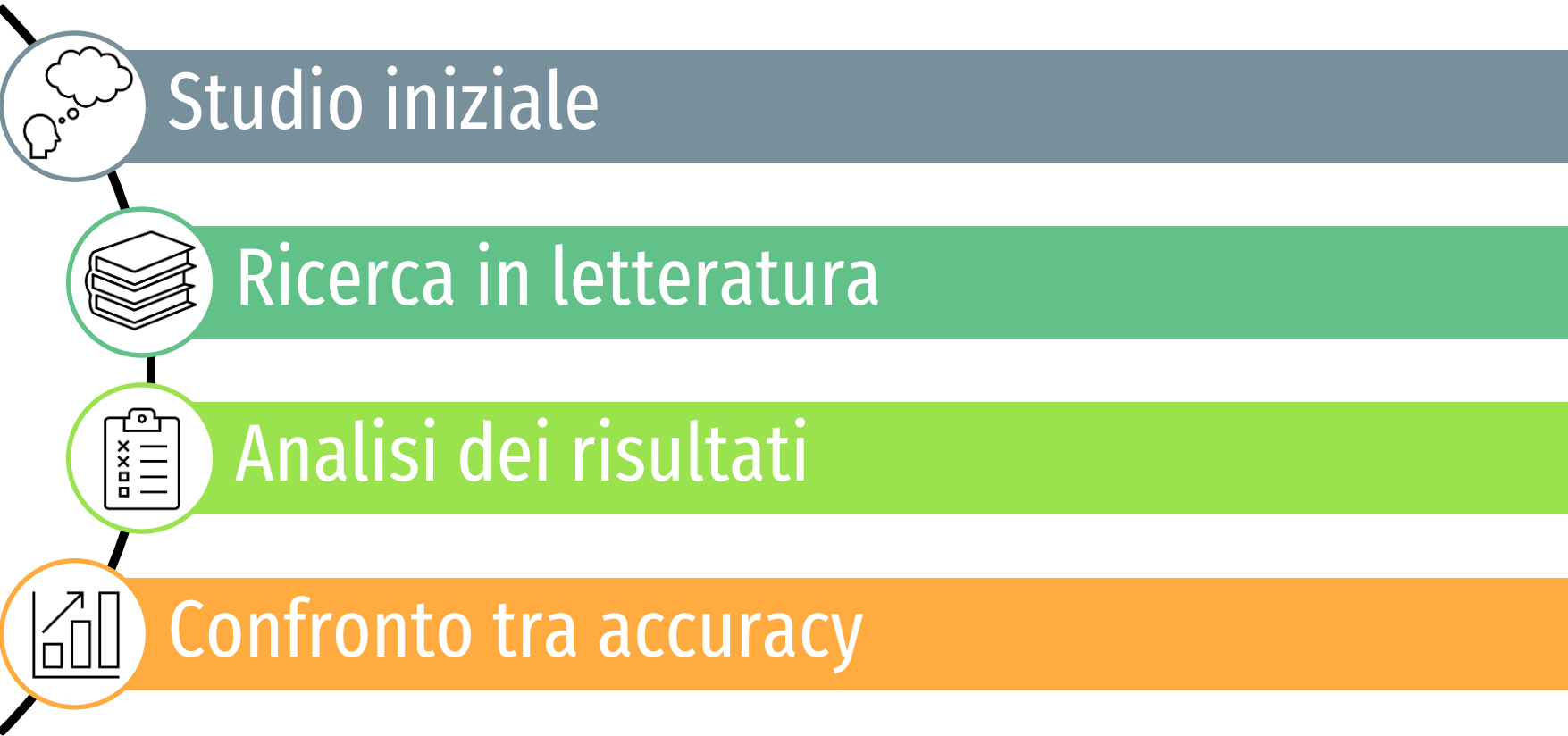
Valutazione classi

● Rimedio classi sbilanciate

- Data augmentation per oversampling su classe 162 ha fallito (20% di f1-score)
- Undersampling non adatto
- **Classi pesate**



Degraded Test Set





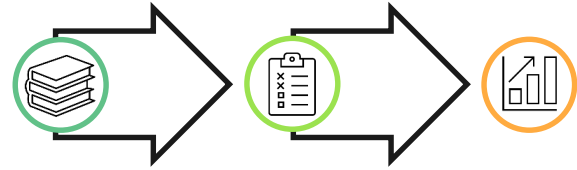
Studio iniziale

- **Assunzioni**

- Limitata diversificazione dei tipi di degradazione

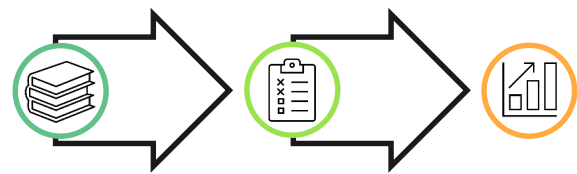
- **Ipotesi**

- Per JPEG discriminare in base al peso
- Incrementare la data augmentation
- Calcolare punteggi di degradazione a ogni immagine

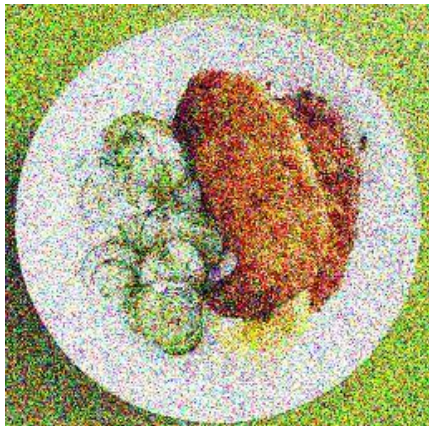




Studio iniziale



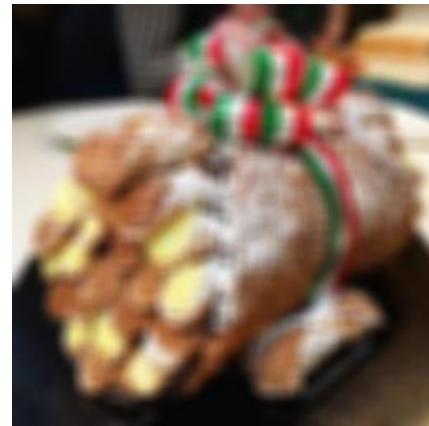
Noise gaussiano



Compressione JPEG

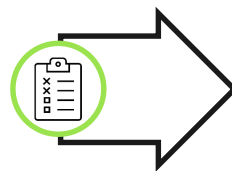


Blur





Ricerca in letteratura



- *Fast Noise Variance Estimation*

- Stima la varianza del rumore gaussiano additivo a media zero
- Quasi insensibile alle strutture in un'immagine
- Filtra l'immagine con N e la confronta con l'originale

L1 =

0	1	0
1	-4	1
0	1	0

L2 =

$\frac{1}{2}$	0	$\frac{1}{2}$
0	-2	0
$\frac{1}{2}$	0	$\frac{1}{2}$

N = 2*(L1 - L2) =

1	-2	1
-2	4	-2
1	-2	1

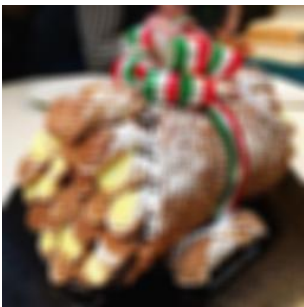


Analisi dei risultati



- Media: 8
- Minimo: 0.06
- Massimo: 49.8

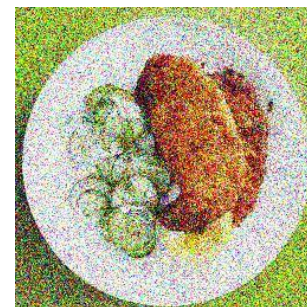
Threshold: 0.41



20% - 2399



Threshold: 27



67% - 8031

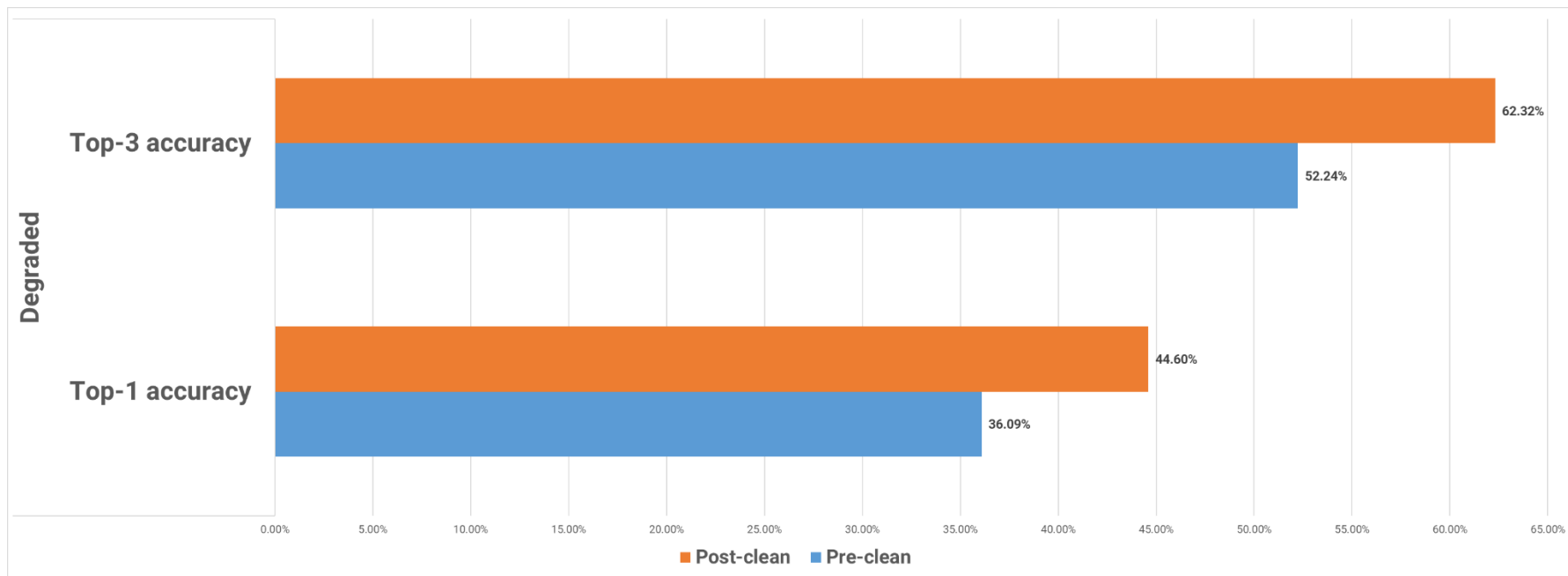


13% - 1564



Confronto tra accuracy

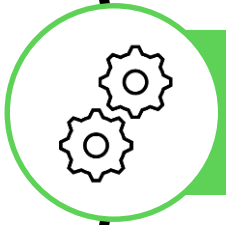
- Confronto performance del modello finale sul degraded test
 - Top-1 accuracy ideale: 50%



Category Search



Studio iniziale



Approccio scelto



Analisi dei risultati



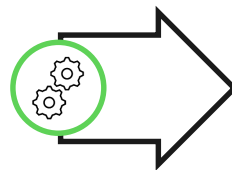
Studio iniziale

● Problemi

- Tempistiche
- Oggettivizzazione dei risultati
 - Pulizia training set
- Similarità tra classi
 - *Concetto di similarità*

● Assunzioni

- Risultati oggettivi da considerarsi *indicativi*
- Possibilità di utilizzo della rete addestrata
- Fase di *pre-elaborazione one shot*



val_011915.jpg, label 243: mostaccioli



train_088538.jpg, label 147: casserole





Approccio scelto

- **Soluzione scelta**
 - Estrazione feature training set con CNN MobileNetV3Large
- **Motivazioni**
 - Tempo di progettazione: **immediato**
 - Tecnica già trattata
 - Tempo computazionale
 - 1h 30m di pre-elaborazione
 - Primi risultati incoraggianti
- **Dilemma**
 - MobileNetV3Large pre-addestrata su
 - Imagenet?
 - FoodX-251?
 - Dove effettuare il taglio?



Immagine query da internet



Primi risultati

Immagine: train_092525.jpg, Label: 198, Punteggio: 78.46

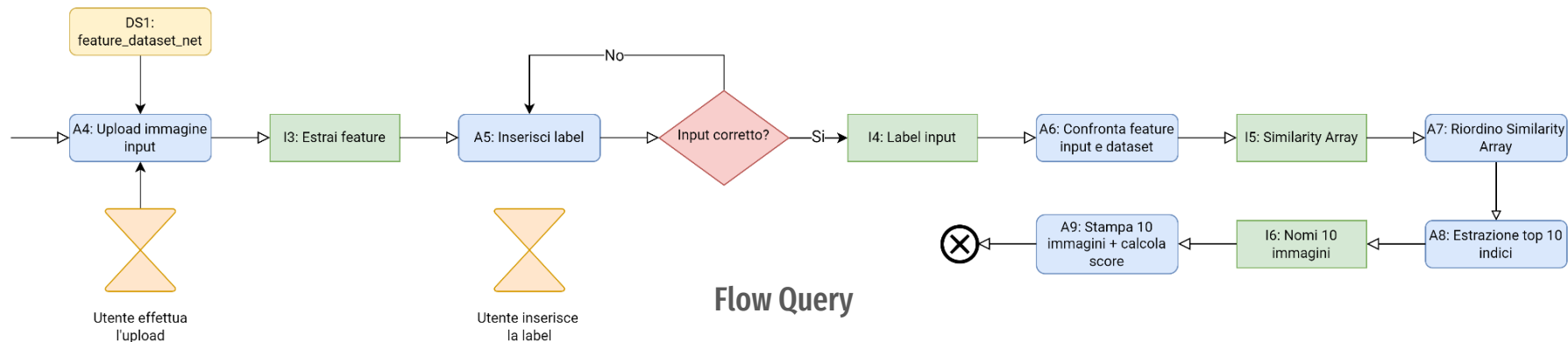
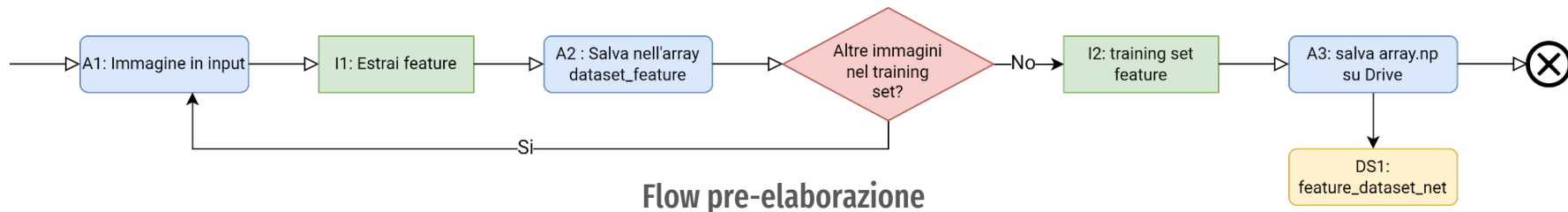


Immagine: train_088410.jpg, Label: 147, Punteggio: 78.44





Approccio scelto





Analisi dei risultati

- Test manuali

- Risultati semi-oggettivi e visualizzazione delle performance

Label
incorretta nel
training

Query



Immagine: train_059570.jpg, Label: 0, Punteggio: 62.83



Immagine: train_059732.jpg, Label: 0, Punteggio: 59.43



Immagine: train_059476.jpg, Label: 0, Punteggio: 58.76



Immagine: train_059763.jpg, Label: 0, Punteggio: 56.31



Immagine: train_059391.jpg, Label: 91, Punteggio: 56.16



Immagine: train_059513.jpg, Label: 0, Punteggio: 55.53





Analisi dei risultati

- **Category search automatizzato**

- Necessario un set di immagini già categorizzate → Test set
- Utile per stimare delle performance generiche

- **Risultati ottenuti**

- Campione utilizzato
 - 1000 immagini test set fornito
- MobileNetV3 fine-tuning su FoodX-251 + taglio Dense_Add_Layer
 - Media label corrette: 32.80%
 - Deviazione standard: 31.76
- MobileNetV3 fine-tuning su FoodX-251 + no taglio
 - Media label corrette: 41.80%
 - Deviazione standard: 41.06
- MobileNetV3 ImageNet
 - Media label corrette: 21.33%
 - Deviazione standard: 23.89

Grazie per l'attenzione!

Classificazione Dataset FoodX-251

Slide Extra

Appendice 1

- GLCM

- Gray-Level Co-occurrence Matrix
- Cattura ripetizioni di coppie di intensità di grigio vicine
- Permette di ottenere
 - Contrasto
 - Correlazione
 - Energia
 - Omogeneità

0	1	0	2
0	2	1	1
3	1	0	0
0	0	2	3

4 x 4 image

2	1	3	0
2	1	0	0
0	1	0	1
0	1	0	0

GLCM

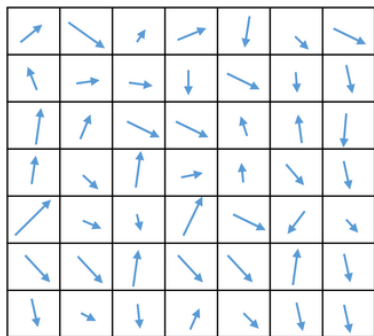
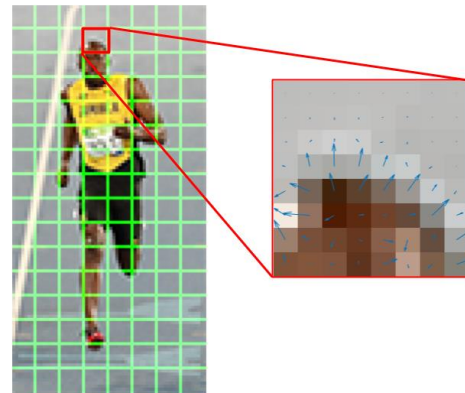
0.16	0.08	0.25	0
0.16	0.08	0	0
0	0.08	0	0.08
0	0.08	0	0

Normalized GLCM

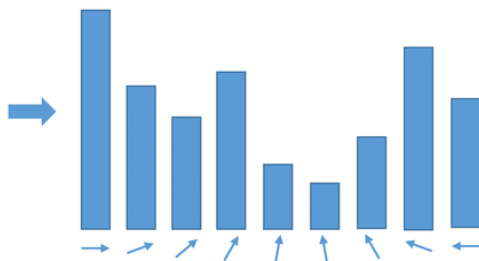
Appendice 2

- HOG

- Histogram of Oriented Gradients
- Per ogni pixel della cella calcola il valore del gradiente e la sua direzione
- Riporta poi questi dati su un istogramma
- Unisce i vari istogrammi, con orientamenti, ottenuti per ogni cella



Gradient magnitude and direction of the cell

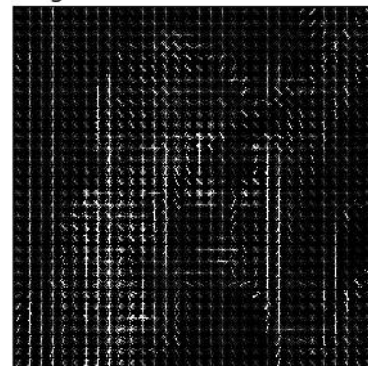


HOG

Input image

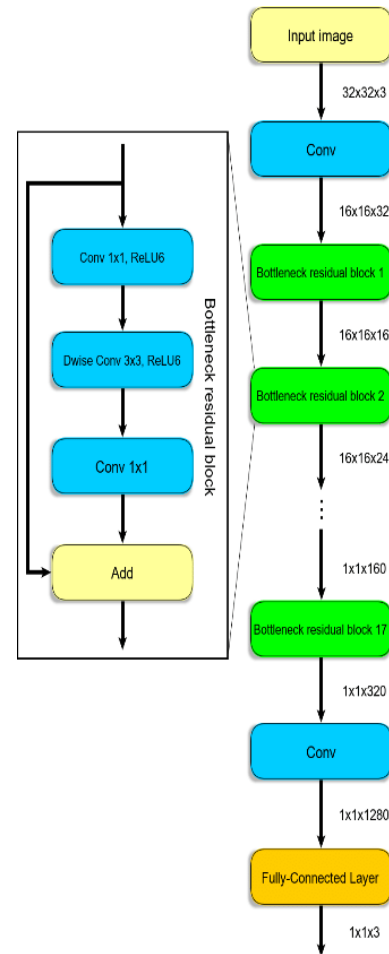


Histogram of Oriented Gradients



Appendice 3

- Ogni blocco contiene uno strato *depthwise separable convolution*
 - Un filtro per canale anziché un filtro per tutti
 - Divide l'operazione di convoluzione in due fasi
 - Convoluzione separata in **profondità**
 - Convoluzione **attraverso i canali**
 - Dalla v2 abbiamo i **bottleneck residual**
 1. **Espandiamo** la dimensionalità dell'input per le depthwise conv
 2. Catturiamo le caratteristiche spaziali con la **depthwise conv**
 3. **Contrazione** con convoluzione pointwise
 - Vanishing gradient problem
 - **Residual** perché dopo la contrazione, aggiungiamo l'input originale all'output
- Generalmente usiamo la ReLU, la v3 introduce l'hard sigmoid
- In uscita, per ridurre il numero di parametri utilizziamo un GAP (Global Average Pooling)



Appendice 4

- **Cosine similarity** utilizzata per misurare la similarità tra due vettori
- Preferita alla distanza Euclidea con vettori grandi

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Appendice 5



Ricerca in *letteratura* del significato di
«POI» ha prodotto scarsi risultati:

